ECE 471 – Embedded Systems
Lecture 27

Vince Weaver

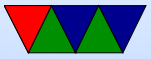
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

14 November 2018

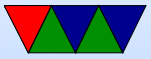
Announcements

- I will respond soon to project ideas!
- Midterm Friday
- Makeup test on Thursday if you need it, contact me please
- HW grades have been posted



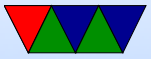
Academic Integrity

- I shouldn't have to be telling this to a 400 level class
- Things that are OK
  - Writing own code
  - Writing code, sharing with partner
  - Writing code, needing help, ask me, or the TA (colin.leary)
  - Writing code, doing a google search for help (help! not looking for code to copy, but for advice)
  - Writing code, asking other students in class to look at



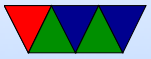
it or discuss at high level the problem

- Things that are *not* OK
  - Taking someone else's code verbatim and turning it in as yours.
  - Cut-and-pasting major chunks of code from someone else, turning it in as yours
  - Taking someone else's code, retyping it, maybe changing the indentation, comments, and copy/replacing a few variable names and turning it in as yours
- We learned in the last few classes why copying code



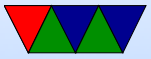
you don't really understand is bad (could read to rocket crashes, etc)

- It's also a good way to get your company sued if you get caught doing this at your job

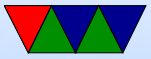


Homework 6 Review

- i2c: No, the clock is not fully implemented in the protocol. What does that mean? Many groups used that exact wording too.
- read_SCL is unused. how would it be used? clock stretching
- Yes – can be used for system load but that's not really the intended purpose. Test interrupts?
- Code: be careful cutting and pasting! A lot of issues where you cut and paste SDA code to SCL but forgot

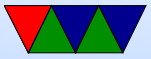


to change SDA to SCL
Changing what GPIOs are used?

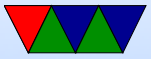


Homework 7 Review

- Be sure to set single-ended mode properly (see data sheet)
  0xa not 0x2
- Make sure for temp you have it hooked to CH2
- How to clear all but bottom 2 bits? and with 0x3 (not 0x2)
- When doing C to F, be sure to use floating point constants 5.0/9.0 or else C does integer math.
  really, think about what temp it is

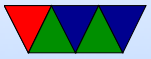


- Use the right conversion routines. The ones in class/the manual, not just something you find on the internet
- Don't memset after you set the values for the ioctl
- people with Analog Discovery



Homework 8 Review

- Some C coding mistakes
  - Error checking. Exit if cannot open. If you don't, can segfault if try to scanf error fd
  - Returning -1 on error might be bad idea
  - Check for errors! What could go wrong? What if different sensor used, don't segfault.
  - If using streams (FILE *fff), on fopen() error it returns NULL, not -1.
  - Be sure to close files, otherwise leak file descriptors

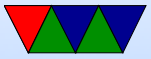


- Confusion about how #define SENSOR works. Can't include C pre-processor directives inside of a string
- fgets only gets up to linefeed
- Finding a file using C. opendir() readdir(), horrible interface

- Why need Vdd? To provide enough current for this particular chip needs extra current if you want parasite mode.

  You can try without Vdd but you will always read out 85C.

  Manual suggests MOSFET, but apparently it's possible

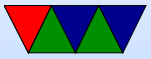


on Pi if use 4.7k resistor as well as “strong-pullup=y” kernel command line option.

- Because of distance, 1-wire
- shell script

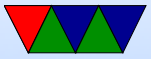
  Trouble if edit on windows, why (linefeed vs carriage return)

  shebang description. Making executable with chmod

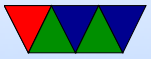


Midterm Review

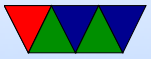
- Booting on the Pi
  - What a bootloader does
  - Why Pi is unusual
- Real Time
  - Definitions
  - Is this hard, soft, firm
- i2c/SPI/1-wire
  - Know the tradeoffs between
  - Be able to follow the C code for them



- Security
  - Buffer overrun, why it is bad
- Coding Practices
  - Be aware of the case studies we suggested
  - Know of some of the recommended ways to write safer C code

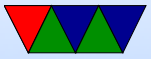


Stuff from last time



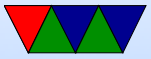
Good Test Practices

- Unit testing
- Test Driven Development – tests written before the code happens, needs to pass the tests before done
- Fuzzing
- Documentation
  Source control



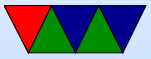
Space Shuttle Design

- https://www.nasa.gov/mission_pages/shuttle/flyo flyfeature_shuttlecomputers.html
- Issues normal embedded systems don't have: Vibration at liftoff, Radiation in Space
- If computer stopped for more than 120ms, shuttle could crash
- "Modern" update in 1991: 1MB Ram, 1.4MIPS. Earlier was 416k and 1/3 as fast and twice as big
- Change to code, 9 months testing in simulator, 6 months



more extensive testing

- 24 years w/o in-orbit SW problem needing patches
- 12 year stretch only 3 SW bugs found
- 400k lines of code
- HAL/S high-order assembly language (high-level language similar to PL/I)
- PASS software – runs tasks. Too big to fit in memory at once
- BFS – backup flight software. Bare minimum to takeoff, stay in orbit, safely land, fits in memory, monitors pASS during takeoff/landing Written by completely different



team.

- 28 months to develop new version
- IBM
- Extensive verification. One internal pass, one external
- 4 computers running PASS, one running BFS
- Single failure mission can continue; still land with two failures
- 4 computers in lock-step, vote, defective one kicked out